

MENU

SEARCH

INDEX

DETAIL

BACK

NEXT

5/7



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 06289875

(43)Date of publication of application:
18.10.1994

(51)Int.Cl.

G10H 1/36

G10H 1/00

G10H 1/40

(21)Application number:
05072365(71)Applicant: KAWAI MUSICAL INSTR MFG
CO LTD

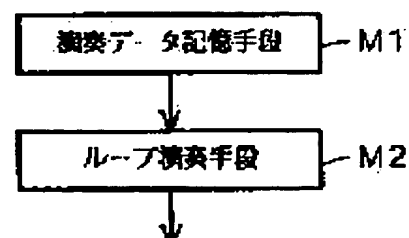
(22)Date of filing: 30.03.1993

(72)Inventor: SAITO TSUTOMU
YAMADA MASAHIKO
MINOURA HARUO
NAKAGAWA HIRONOBU

(54) AUTOMATIC MUSICAL PLAYING DEVICE

(57)Abstract:

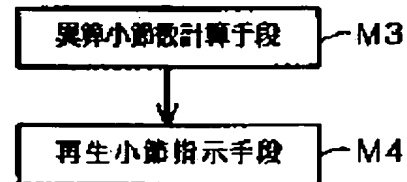
PURPOSE: To suppress a waste of an automatic musical performance memory and to smoothly reproduce an accompaniment even when rhythm styles are switched by providing a loop playing means, a cumulative measure quantity calculating means, and a reproduced measure indicating means.



BLANK PAGE

BLANK PAGE

CONSTITUTION: The loop playing means M2 repeatedly reads out performance data on loop measures and carries out a musical performance. The cumulative measure quantity calculating means M3 accumulates the number of played measures from the start of the musical performance and the reproduced measure indicating means M4 indicates reproduced measures of the musical performance data on the basis of the remainder obtained by dividing the accumulated number of measures by each number of loop measures. Namely, the remainders obtained by dividing the cumulative number of measures of the musical performance data played in a loop by the respective different numbers of measures indicate the positions of reproduced measures. Consequently, even when the number of loop measures are different, a measure to be played can be specified immediately, so a musical performance which gives no feeling of physical disorder can be carried out, for example, even when rhythm styles are switched or when one part is reproduced from a halfway point.



LEGAL STATUS

[Date of request for examination]	06.11.1996
[Date of sending the examiner's decision of rejection]	05.10.1999
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	
[Date of registration]	
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

Copyright (C); 1998 Japanese Patent Office

[MENU](#)[SEARCH](#)[INDEX](#)[DETAIL](#)[BACK](#)[NEXT](#)

BLANK PAGE

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-289875

(43)公開日 平成6年(1994)10月18日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 1 0 H 1/36		4236-5H		
1/00	1 0 2 Z	4236-5H		
1/40		4236-5H		

審査請求 未請求 請求項の数 2 O L (全 21 頁)

(21)出願番号 特願平5-72365

(22)出願日 平成5年(1993)3月30日

(71)出願人 000001410

株式会社河合楽器製作所
静岡県浜松市寺島町200番地

(72)発明者 斉藤 勉

静岡県浜松市寺島町200番地 株式会社河合楽器製作所内

(72)発明者 山田 雅彦

静岡県浜松市寺島町200番地 株式会社河合楽器製作所内

(72)発明者 箕浦 晴夫

静岡県浜松市寺島町200番地 株式会社河合楽器製作所内

(74)代理人 弁理士 足立 勉

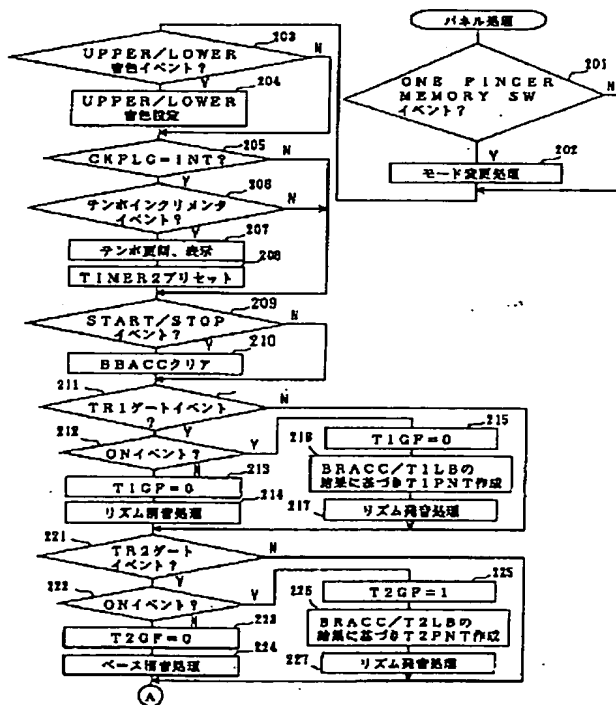
最終頁に続く

(54)【発明の名称】 自動演奏装置

(57)【要約】

【目的】 自動伴奏メモリの無駄を抑えるとともに、リズムスタイルが切り換えられた場合でも伴奏の再生を円滑に行なうことができる自動演奏装置を提供すること。

【構成】 ステップ216では、異なるループ小節数を有するトラックに変更されたので、次に伴奏される小節を決める処理が行われる。つまり、累算小節数BBACCをTR1のループ小節数T1LBで割った余りの小節数を求め、この余りの小節数に基づいて現在のTR1ポインタT1PNTを設定する。即ち余りの小節数に対応して定まるBEAT数を、現在のTR1ポインタT1PNTとする。これによって、ループ小節数が異なるトラックに切り換えられた場合でも、容易に次に伴奏すべき小節を求めることができる。



【特許請求の範囲】

【請求項1】 ループ小節数の異なる複数の演奏データを記憶する演奏データ記憶手段と、
該演奏データ記憶手段に記憶された複数の演奏データを、繰り返して読み出して演奏を行なうループ演奏手段と、

該ループ演奏手段による演奏の開始から、演奏される小節数を累算する累算小節数計算手段と、

該累算小節数計算手段によって累算された累算小節数を、前記異なるループ小節数で除した余りに基づいて前記演奏データの再生小節を指示する再生小節指示手段と、

を備えたことを特徴とする自動演奏装置。

【請求項2】 前記各ループ小節数を2の累乗とし、2進数で表示される累算小節数の所定ビットをマスクすることにより、演奏の再生小節を指示することを特徴とする前記請求項1記載の自動演奏装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、所定のループ小節における伴奏パートを、繰り返して自動演奏する自動演奏装置に関する。

【0002】

【従来の技術】従来より、電子オルガンや電子キーボード等の自動演奏装置においては、例えば自動リズム、自動ベース、自動アカンパニメントの様な伴奏パートの自動再生を行なう自動伴奏機能を備えているものが知られている。

【0003】この自動伴奏を行なうために、自動伴奏メモリ内に、通常はリズム、ベース、アカンパニメントの演奏内容を、同一小節数分記憶している。具体的には、図18に示す様に、リズムR、ベースB、アカンパニメントA共通の1つのシーケンストラックに、各伴奏パートの音高（音色）や発音タイミングを混合して記憶している。

【0004】

【発明が解決しようとする課題】ところが、上述した技術では、次の様な問題があり、必ずしも好適ではなかった。即ち、図18の様に、自動伴奏メモリである共通のシーケンストラックに、リズムR、ベースB、アカンパニメントAの伴奏データを併せて記憶するものでは、自動伴奏メモリをアクセスするポインタは1つで済むが、リズムR、ベースB、アカンパニメントAの最小公倍数的な（繰り返して演奏される）ループ小節数を必要とするという問題があった。

【0005】例えばリズムRは2小節、ベースBは4小節、アカンパニメントAは1小節でループさせたい場合には、全ての伴奏パートを4小節分記憶しなければならず、図18の斜線で示す小節が他の小節の内容と重複する（A1～A4は同一内容となる）ので、メモリに無駄

が多かった。

【0006】特に、近年では、演奏の高品質化に伴ってループ小節数が多くなる傾向にあり、このメモリの無駄という問題は一層大きな問題となっている。この対策として、例えば各伴奏パート独立に各々トラックを設け、ループ小節を記憶し再生する処理が考えられるが、単純にその様に構成すると、途中でリズムスタイルが切り換えられた時などには、各々のパートは新たなリズムスタイルのどの部分からスタートしてよいか分からないという別な問題が生じてしまう。

【0007】特に電子オルガンにおいては、伴奏中にリズムスタイルが切り換えられると、通常その続きから伴奏が行われるので、この点は大きな問題である。本発明は、前記課題を解決するためになされ、自動演奏メモリの無駄を抑えとともに、リズムスタイル等が切り換えられた場合でも、伴奏の再生を円滑に行なうことができる自動演奏装置を提供することを目的とする。

【0008】

【課題を解決するための手段】この目的を達成するための請求項1の発明は、図1に例示する様に、ループ小節数の異なる複数の演奏データを記憶する演奏データ記憶手段M1と、該演奏データ記憶手段M1に記憶された複数の演奏データを、繰り返して読み出して演奏を行なうループ演奏手段M2と、該ループ演奏手段M2による演奏の開始から、演奏される小節数を累算する累算小節数計算手段M3と、該累算小節数計算手段M3によって累算された累算小節数を、前記異なるループ小節数で除した余りに基づいて前記演奏データの再生小節を指示する再生小節指示手段M4と、を備えたことを特徴とする自動演奏装置を要旨とする。

【0009】また、請求項2の発明は、前記各ループ小節数を2の累乗とし、2進数で表示される累算小節数の所定ビットをマスクすることにより、演奏の再生小節を指示することを特徴とする前記請求項1記載の自動演奏装置を要旨とする。

【0010】①ここで、前記自動演奏装置としては、例えば電子オルガン、電子ピアノ、電子キーボード等が挙げられる。

②本発明は、MIDIを用いてキーボードから入力した信号を、パーソナルコンピュータ等の汎用コンピュータ上で処理して自動演奏を行なうものにも適用できる。

【0011】③前記演奏データ記憶手段に記憶される異なるループ小節数とは、少なくとも二つ以上の異なるループ小節数を意味し、全てのループ小節数が異なるとは限らない。

④前記複数の演奏データは、（後述する実施例の様に）各々異なるトラックに区別して記憶すると、従来の同一ループ小節数だけ記憶する方式に比べて少ないメモリで済み、しかも読み出し時に不要な演奏情報（1つのトラックに複数パートの演奏情報を記憶して読み出す際には

対象パート以外は不要な演奏情報となる)がないので、CPUが効率よく処理できる。

【0012】また、例えばループ小節数の異なる複数の演奏データを1つのトラックに記憶することも可能であり、その場合には、上述の記憶方法(別トラックに同一ループ小節数を記憶)以上に圧縮して記憶できる。但し、この場合には、ループ小節数の違う演奏情報は読み出しても捨てなければならないので、CPUは読み出し時には不要な演奏情報を廃棄処理する時間は浪費することになる。

【0013】更に、メモリ効率とCPUの処理効率が最も良いのは、同じループ小節数の演奏情報を1つのトラックに複数記憶したいいくつかのループ小節数毎のトラックとしてメモリに記憶することである。こうすることにより、読み出したポイントも全部ではなく各トラック単位で用意すればよく、演奏情報の記憶に関しても同一ループの複数の演奏情報で1小節に1つの小節マークのみ記憶すればよいため、無駄がない。

【0014】⑤前記複数の演奏データを読み出す場合には、同時に、各々の演奏データを読み出す場合が一般的であるが、この同時とは、各演奏データをポイント等を用いて同様なタイミングで読み出すことを意味する。尚、CPUの処理として、時分割で処理する場合も、この同時の概念に入る。

【0015】⑥前記異なるループ小節数で除した余りに基づいて演奏データの再生小節を指示する再生小節指示手段としては、実際に累算小節を除することも考えられるが、例えば累算小節数及びループ小節数に対応した余りが記憶されているテーブルを参照することによって、読み出すべき小節数を指示することもできる。

【0016】⑦また、前記2進数で表示される累算小節数の所定ビット(例えば上位ビット)をマスクすると、具体的には、例えば累算小節数の8ビットの2進カウンタとマスク用に設定された8ビットの2進データの論理積を求めることによって実現できる。

【0017】

【作用】

(1)請求項1の発明では、演奏データ記憶手段M1によって、ループ小節数の異なる複数の演奏データを記憶するとともに、ループ演奏手段M2によって、このループ小節の演奏データを繰り返して読み出して演奏を行なう。そして、累算小節数計算手段M3によって、演奏の開始からの演奏される小節数の累算を行ない、再生小節指示手段M4によって、この累算された累算小節数を各ループ小節数で除した余りに基づいて演奏データの再生小節を指示する。

【0018】つまり、ループして演奏される演奏データの累積小節数を各々異なるループ小節数で割った余りが、再生される小節の位置を示すことになる。これにより、ループ小節数が異なっている場合でも、即座に際す

べき小節を指定できるので、例えばリズムスタイルが切り換えられた場合や(最初は再生されなくて)途中からあるパートが再生される場合でも、違和感のない演奏を行なうことが可能である。

【0019】しかも、この様に、例えば各リズムスタイルや各トラック毎に、記憶するループ小節数を違えることが可能で、最小のループ小節数を設定することができるので、演奏の質を低下させることなく、メモリを低減することが可能である。

(2)特に、請求項2の発明では、各ループ小節数を2の累乗とし、2進数で表示される累算小節数の所定ビットをマスクすると、一層容易に再生小節が分かるので好適である。

【0020】ここで、マスクすることによって、次に再生すべき(読み出すべき)小節数を決める手順について、詳細に説明する。パネルに10進数で表示される累算小節数と、それを2進数で積算して記憶するBARカウンタとの間には、下記表1に示す関係がある。

【0021】

【表1】

パネル表示累算小節数 10進3桁	BARカウンタ 2進8桁
第001小節	0 0 0 0 0 0 0 0
第002小節	0 0 0 0 0 0 0 1
...	...
第014小節	0 0 0 0 1 1 0 1
...	...
第030小節	0 0 0 1 1 1 0 1
...	...
第099小節	0 1 1 0 0 1 0 1
第100小節	0 1 1 0 0 1 1 0
...	...
第256小節	1 1 1 1 1 1 1 1

【0022】また、ループ小節数が2の累乗に設定されている場合は、(読み出すべき小節数を決めるための)マスクデータは、下記表2の様に設定されている。

【0023】

【表2】

マスクデータ 0 0 0 0 M3 M2 M1 M0	ループ小節数
0 0 0 0 0 0 0 0	1小節ループ
0 0 0 0 0 0 0 1	2小節ループ
0 0 0 0 0 0 1 1	4小節ループ
0 0 0 0 0 1 1 1	8小節ループ
0 0 0 0 1 1 1 1	16小節ループ

【0024】従って、各伴奏において次に演奏すべき小節は、タイマ割込等で作成されたBARカウンタと前記マスクデータとの論理積で求められることになる。例えば4小節ループの伴奏において、(累算小節数を示す)パネル表示が第30小節であるときには、BARカウンタは00011101bであるので、このBARカウンタと第4小節ループ(00000011b)との論理積を求めると0000001bとなり、これは第2小節の中をプレイしていることを示す。ここで、リズムスタイルを切り換えるスイッチが

操作されて、例えば16小節ループの伴奏に切り換えられた場合には、BARカウンタ(00011101b)と第16小節ループ(00001111b)との論理積を求めると、第14小節(00001111b)を示すものとなるので、即時第14小節の途中から再生を行なうことができる。

【0025】つまり、BARカウンタに対し、ループ小節数に応じたマスクを施すことによって、即時にループ小節数に好適に対応した再生小節を求めることができることになる。

【0026】

【実施例】以上説明した本発明の構成・作用を一層明らかにするために、以下本発明の自動演奏装置の好適な実施例について説明する。図2は、実施例としての自動演奏装置の電氣的構成を示し、図3はその操作パネルを示している。

【0027】本実施例の自動演奏装置は、複数の鍵盤(キーボード)を備えた電子オルガンであり、図2に示す様に、この電子オルガンでは、CPU1、ROMA3、ROMB5、RAM7、楽音発生器9、サウンドシステム11、操作パネル13、キーボード15、MIDI入力部17、タイマー19が、バスライン21等を介して接続されている。

【0028】以下、各構成について詳細に説明する。CPU1は、テンポの周期の演算処理や設定されたテンポでの伴奏等の各種の演算処理を行なう。このCPU1は、外付け水晶23により発振し、それを分周して得られたクロック信号はタイマー19に送られる。

【0029】ROMA3は、CPU1で行なう演算のために、各種の制御プログラムを格納している。ROMB5は、ワルツ、スイング等の8つのリズムスタイル毎に、パート1(リズム)、パート2(ベース)、パート3(ACC:アカンパニメント)の自動伴奏データを格納しており、各自動伴奏データは、イントロ、フィルイン、メイン、エンディングの4つのパターンからなる。

【0030】RAM7は、CPU1が各種処理を行なう時のワーキングに使用する。尚、後述する図9に示す各種レジスタもRAM7の一部を利用する。楽音発生器9は、32チャンネル(CH)時分割にて楽音を発生し、そのうち16CHをマニュアル演奏楽音、残り16CHを自動伴奏楽音に使用する。

【0031】サウンドシステム11は、楽音発生器9から送られるデジタル波形信号をD/Aして増幅し、図示しないスピーカから発音する。操作パネル13には、図3に示す様に、マニュアルにてテンポを設定(内部設定)するテンポ操作子24、設定されたテンポを表示するテンポ表示器25、各種のスイッチ(以下スイッチをSWと記す)及び各SWに対応して配置されたLED37が設けられている。そして、この操作パネル13からマニュアルによって指示された内容がCPU1等へ送られる。

【0032】前記SWとしては、伴奏の開始及び停止を指示するスタート/ストップ(START/STOP)SW26、伴奏の途中で所定の伴奏を挿入するフィルイン(FILL IN)SW27、伴奏の開始時及び終了時に所定の伴奏を行なうイントロ/エンディング(INT/END)SW28、伴奏された小節数を積算して表示する累算小節表示器29、テンポの内部設定及び外部設定を切り換えるクロック切換SW31、3種のループ小節を各々記憶するトラックを指定するトラック1, 2, 3(TR1~TR3)のゲートSW32、8種類のリズムスタイルを選択するリズムスタイルSW33、キーオフ後も伴奏データを記憶するためのメモリ(MEMORY)SW34、ワンフィンガーによって和音を指定するためのワンフィンガー(ONE FINGER)SW35等のSWが設けられている。

【0033】図2に戻り、キーボード15は、上(UPPER)鍵盤15a及び下(LOWER)鍵盤15bからなり、このキーボード15には各鍵の動きを検知する図示しないSWアレイよりなるSWが設けられている。そして、キーボード15からマニュアルによって指示された内容(伴奏及び操作を示す信号)がCPU1等へ送られる。

【0034】MIDI入力部17は、外部装置からのMIDI入力信号を受け、パラレル変換して、後に図16にて示すMIDI入力割込み(INT1)をCPU1にかける。タイマー19は、2つのプリセットブルダウンカウンタ(TIMER1, 2)で構成され、その1つ(TIMER2)は、操作パネル13のテンポ操作子24の操作に対応した値をプリセットすることによりダウンカウントして、ゼロになると、後に図17にて示すTIMER2テンポ割込み(INT2)をCPU1にかける。

【0035】もう1つ(TIMER1)は、MIDI入力のタイミングクロック(以下クロックをCKと記す)が入力された時にクリアされ、次のタイミングCKが入力されるまでの時間(周期)を測定するのに使用される。次に、電子オルガンのROMB5に記憶された各自動伴奏データのヘッダーについて、図4に基づいて説明する。

【0036】リズムスタイルは、ワルツ(WALTZ)、スイング(SWING)、バラード(BALLAD)、タンゴ(TANGO)、ラテン(LATIN)、サンバ(SAMBA)、8ビート(8BEAT)、16ビート(16BEAT)の8種類である。それらは各々リズムトラック(以下トラックをTRと記す)(TR1)、ベースTR(TR2)、ACCTR(TR3)からなり、各TR(TR1, TR2, TR3)の中は、イントロパターン、メインパターン、フィルインパターン、エンディングパターンから構成される。

【0037】このうちイントロパターンは、自動伴奏停

止中にINT/END SW28が操作され、対応したLED37が点灯している時に、START/STOP SW26を操作して、ランニング（自動伴奏再生）状態に入った時に、最初のi小節（i=1, 2, 4のいずれか）を1回限り再生するパターンであり、このパターン長はリズム、ベース、ACC共通である。

【0038】メインパターンは、自動伴奏再生中のノーマルパターンであり、繰り返し再生される。この部分は、伴奏の質の向上のために、伴奏のリズムスタイル毎にそのパターン長（ループ小節数）が異なる様に設定されて、しかもリズム、ベース、ACCによってもそのループ小節数が異なる様に設定されている。また、例えば図5に示す様に、あるリズムスタイル（例えばワルツ）では、リズムTR（TR1）は4小節でループし、ベースTR（TR2）は2小節でループし、ACC TR（TR3）は8小節でループするといった具合に、ループ小節数が2の累乗となる様に設定されている。

【0039】フィルインパターンは、メインパターン進行中に操作パネル13のFILLIN SW27が押された時、f小節（f=1, 2, 4のいずれか）を1回限り再生するパターンであり、このパターン長（ループ小節数）はリズム、ベース、ACC共通である。尚、フィルインパターン終了後、メインパターンはフィルイン割込みなしの場合に再生しているべき小節位置へジャンプする。

【0040】エンディングパターンは、自動伴奏再生中にINT/END SW28が操作され、対応したLED37の点灯している時に、START/STOP SW26を操作して、停止状態に入った時、最後のe小節（e=1, 2, 4のいずれか）を1回限り再生するパターンであり、このループ小節数はリズム、ベース、ACC共通である。

【0041】また、前記各パターンのデータは、図4に示す様に、対応パターンデータの先頭番地である「スタートアドレス」、そのパターンの全イベント数（小節マークも含む）を表わす「イベントバリュー」、そのパターンの小節数を表わす「パターンレングス」、パターンの初期音色/音量を表わす「トーンナンバ（TONE No.）/ボリューム（VOLUME）」からなる。尚、トーンナンバ/ボリュームは、後述するパートA、B共通である。

【0042】次に、前記各パターンにおける1小節内のデータの並びを、図6に表わす。まず、小節マークがあり、それに続いて1小節分の音符（ノート）A～Nが並ぶ音符データ形式であるが、これは、リズムパターン、ベースパターン、ACCパターンも同じ形式である。従って、リズムパターンもベース再生時に読出せば、音色設定次第でそれなりのオートベースになるし、その逆も言える。これによって、リズム、ベース、ACCという異なる自動伴奏パターンデータを全く共通に管理でき

る。

【0043】次に、各伴奏データ及び小節マークの構造を図7に示すが、ここでは全て4バイト構成となっている。

(a) ノートデータは、発音開始時間を示すステップタイムSTEP TIME（前小節からのBEAT数）、音高を示すノートNOTE、音長を示すゲートタイムGATE TIME（キーオン時間に相当するBEAT数）、音の強さ（タッチ）を示すベロシティVELOCITY、パートゲイト（PA：パートA、PB：パートB）からなる。尚、このパートとは、和音種類群（例えばメジャー系PA/マイナー系PB）を意味し、現在例えばLOWERキーの押鍵から検出されたコードタイプがどちらのパート（PA/PB）に属するかを判断し、対応するパート側音符情報のみを再生発音するためのものである。これにより、従来よりも効率よく和音種類群毎に独立した符長をもつパターンを記憶できる。

【0044】(b) LED表示データは、テンポに同期したリズムスタイル固有の点滅パターンを記憶再生するもので、リズムTRのみに含むことができる。ここでのゲートタイムは点灯時間を表し、L0～L7は8つのLEDを表わす。

(c) 音色データは、パターン開始直後の初期音色に関してはヘッダーに記憶されており、パターン再生途中で音色変更する時のみこの音色データフォーマットを使用する。尚、音色はパートA、B共通である。

【0045】(d) 音量データも、音色同様パターン再生途中で音量変更する時のみこの音量データフォーマットを使用する。

(e) オートベンド（BEND）データは、図8（a）及び下記表3に示す様に、発音中の音符を一旦所定ピッチまでずらした後、設定時間をかけて徐々に基本ピッチに戻すものであり、パートA、B独立に設定できる。ここで、セント（cent）とは、基準ピッチからのずれ具合を意味する。

【0046】このオートベンドデータは、図8（b）に示す様に、±（UP/DOWN）、Depth（ベンドの深さ）、Release Time（戻るのにかかる時間）を自由に設定することで任意のオートベンドを作成できる。尚、アタック（ベンド開始からベンドMaxまで）の時間は、固定で0.1秒とするが、変更可能とすることもできる。

【0047】

【表3】

深さ	cent	回復時間	BEAT
0	20	0	4
1	40	1	8
2	60	2	16
3	80	3	32
4	100	4	48
5	140	5	64
6	170
7	200
		15	192

【0048】このオートベンドデータを採用することで、ピッチ ベンド (PITCH BEND) データとして、アナログ的に多量のイベントを記憶しなくて済む。また、図7のオートベンドデータの2バイト目のbit0のV=1の場合は、その音符のタッチデータVELOCITYとベンドの深さDEPTHを次のように乗算し、タッチによってオートベンドの深さを変えることができる。

【0049】

$DEPTH = DEPTH \times VELOCITY / 64$

但し、VELOCITY=1~127

同様にノートを用いてキースケールで (即ちキーボードのキーの種類で)、深さ及び回復時間を変更することも容易である。

【0050】(f) 小節マークに関しては、そのSTEP TIME=111111**Bの場合が、小節マークであることを示している。つまり、スタート小節マーク111101B, 途中小節マーク11111110B, エンド小節マーク11111111Bとして使い分けることで、小節マークを見ただけでCPU1はスタート/途中/エンドのいずれかを判断できるため、次のパターンの読みだし制御 (イントロからメインへ、メインのループ、フィルインからメインへ、エンディング終了から停止等) が容易になる。

【0051】2バイト目のEVENT VALUEは、その小節内 (次の小節まで) の各種データの総量意味し、この値に4を乗算することで次の小節マークを即座に読出すことができる。3バイト目のTONE No. や4バイト目のVOLUMEは、その小節開始時の音色ナンバと音量を意味する。

【0052】これら小節マーク内のEVENT VALUE, TONE, VOLUMEは、フィルイン挿入後にメインパターンの対応小節へ復起する場合や、メインパターン再生中に突然リズムスタイルが変更され、新たなリズムスタイルの途中から再生する必要があるときに有効であり、処理が高速にできる。

【0053】次に、前記CPU1の使用するRAM7に設定された各種レジスタ (メモリ) を、図9に示す。

(1) 選択リズムスタイル (RYMSTL) メモリは、操作パネル13のリズムスタイルSW33の切り換えによって選択されるリズムスタイルの種類を記憶し、8リズ

ムスタイルの中で現在選択されているもののNo.を格納する。

【0054】(2) 累算小節数 (BBACC) メモリは、小節データ (BAR) 2バイトとビートデータ (BEAT) 1バイトからなり (図10 (a) 参照)、自動伴奏がスタートした時点からの累算小節数とビート数を2進数で記憶する。

(3) スタート/ストップフラグ (SSFLG) メモリは、自動伴奏の状態を示すフラグを記憶し、そのフラグは図10 (b) に示す様に、ビット0, 1にて、00 (イントロ), 01 (メイン), 10 (フィルイン), 11 (エンディング) を示し、ビット7にて、S/S=1 (伴奏中), S/S=0 (停止) を示す。

【0055】(4) TEMPO (テンポスピード) メモリは、現在のテンポスピードを格納し、例えば4分音符/120のときは2進数で120を記憶する。

(5) LOWERキー (LKEY) メモリは、LOWER鍵盤15bに関する情報を記憶するためのメモリであり、8キー×2バイトの容量をもち、各1キーは図10 (c) に示す2バイトのデータを持つ。

【0056】このLKEYメモリのうち、CH NUMBERは、0001bが (LOWERキーの) マニュアル操作を意味し、1001bが (LOWERキーの) MIDI信号による操作を意味する。尚、このLKEYメモリに格納されるキー情報はLOWERキーに限る。

【0057】ONEFGフラグは、ワンフィンガーモード (1本の指でコードタイプ、ルートを指定する) にて付加された音符であるときに、ONEFGフラグ=1となる。DELAYフラグは、DELAY=1の時、LOWERキーイベントがあつてから時間がたっていないので、コード検出のためにまだそのキー情報を使うべきでないことを意味する。

【0058】ON/OFFフラグは、LOWERキーのオン/オフを意味する。つまり、操作パネル13上のMEMORY SW34が操作され、メモリモードになったときはLOWERキーオフ後もこのLKEYメモリにキー情報が残っている。その時ON/OFFフラグが0 (オフ) となる。

【0059】尚、LKEYメモリの内容は前述の毎く、コード検出に利用されるばかりでなく、ACCTRの伴奏パターンを発音する音高情報としても利用される。このACCTRの自動伴奏とは、LOWERで現在押されている音高 (LKEYメモリの音高) を、パターンに記憶されたタイミングとベロシティにて複数同時に刻む方式であり、その最大同時発音数は8音である。

【0060】(6) 図9に戻り、ディレイタイム (DLY TIM) メモリは、コードの検出禁止期間であるDelay Timeを記憶するものである。このDelay Timeの値はコードの検出禁止期間の長さを示し、この検出禁止期間の長さは、下記表4の様に示す様に、ビート (BEA

T) 数にて規定されている。

【0061】このビートは、分解能が4分音符/96とされているので、テンポが4分音符/120のとき、60秒/120/96ビート=5.2ms/ビートとなる。このことから明かな様に、Delay Timeの設定時間、即ちコードの検出禁止期間は、本実施例ではテンポスピードに比例する様に設定されている。勿論、テンポがあまり遅くなった時や、あまりに早くなった時には、絶対時間と比較して一定範囲におさまるようにした方がよい。

【0062】尚、前記DELAYフラグは、LOWERキーイベントが発生した時、LKEYメモリにDELAYフラグ=1として書込まれる。そして、Delay Timeの値はタイマー変化に合わせてCPUメインルーチンで減算(-1)され、この減算によってゼロになると、対応するLKEYメモリのDELAYフラグ=0とされる。即ち、DELAYフラグ=0となると、コードの検出が開始される。

【0063】

【表4】

キー ON/OFF	条 件		Delay Time (ビート数)
	MEMORY モード	ONEFG モード	
オン	0	0	02
オン	0	1	03
オン	1	0	03
オン	1	1	04
オフ	0	0	08
オフ	0	1	0A
オフ	1	0	0C
オフ	1	1	0E

【0064】(7)コードルート(CROOT)メモリは、コード(和音)の根音を記憶し、C(00h), C

#(01h) ... B(0Bh)等として処理する。

(8)コードタイプ(CTYPE)メモリは、コード16種類(例えばMajorminor 7th...)を4ビットで記憶する。

【0065】(9)パート指定(PART)メモリは、前記パートA, Bの指定を行なう。

(10)テンポCKフラグ(CKFLG)メモリは、テンポを内部又は外部のどちらの設定で行なうかを示すメモリであり、テンポ操作子24で指定されたテンポを採用するか(内部設定か)、或はMIDI入力のタイミングCK周期から算出したテンポを採用するか(外部設定か)を選択するフラグを記憶する。尚、この選択は操作パネル13のクロック切換SW31に依存する。

【0066】(11)クロックカウント(CKCNT)メモリは、MIDI入力のタイミングCK間の時間を最新の4つ分だけ格納する。

(12)クロック新平均(CKAVRN)メモリは、CKCNTメモリの最新の4つ分の時間平均を記憶する。

【0067】(13)クロック旧平均(CKAVRO)メモリは、前回まで(即ち現在使用されている)の4つ分の時間平均のデータを記憶する。尚、前記CKFLG=EXT(外部設定)のとき、CKAVROメモリに記憶された値は、TIMER2のプリセット値算出に使用される。また、前記CKAVRNの値は、CKAVROの値と一定以上の差が生じたとき、CKAVROの値をCKAVRNの値に更新する。

【0068】(14)また、TR1(リズム)は、下記表5に示す次のレジスタを持つ。

【0069】

【表5】

ループ小節数 T1LB	ヘッダーに記憶されているパターンレングスの値を記憶 (何小節ループであるかを示す)
Gateフラグ T1GF	操作パネルのTR1 SW32の操作で制御される (このフラグ=1のときのみ再生される)
ポインタ T1PNT	ROMB5に記憶されている伴奏データの読出アドレス (累算小節数に合わせて移動する)
ステップデータ T1STP	次に読出すべきタイミングステップを格納 (累算小節のBEATデータと比較される)
プレイ小節 T1PB	伴奏データの何小節目を再生しているかを記憶 (4小節ループならば0, 1, 2, 3のいずれかである)
ゲートタイム T1GTE	リズム音符のオン時間を示す (同時6音まで再生可能なため8バイトを有する) (リズム発音時間に対応したBEAT数を記憶する)

【0070】尚、TR2(ベース), TR3(ACC)もTR1(リズム)同様のレジスタを持つ。但し、ベースの同時発音数は2音、ACCの同時発音数は6音であ

るが、同時に刻むのでゲートタイムは1バイトである。次に、上述した構成を備えた本実施例の電子オルガンの制御処理について、図11~図17のフローチャートに

基づいて説明する。このうち、図11はメインルーチンでの処理を示し、図12及び図13はパネル処理を示し、図14及び図15は自動伴奏の発音/消音処理を示し、図16はMIDI入力の割込処理を示し、図17はTIMER2のテンポ割込処理を示している。

【0071】まず、図11に基づいて、制御全体のメインルーチンについて説明する。ステップ100にて、電源投入後の初期化を行ない、パネル表示、各種レジスタ、音源パラメータ等のクリアを行なう。ステップ101にて、操作パネル13に設けられた各種のSWを走査し、続くステップ102にて、このパネルSWからのイベントの有無を判定する。ここでイベントがあれば、後述するステップ200のパネル処理を行ない、一方イベントが無ければ、そのままステップ103に進む。

【0072】ステップ103では、UPPERキーボード15aのSWを走査し、続くステップ104にて、このキーボードSWのイベントの有無を判定する。ここでイベントがあれば、ステップ105のUPPERマニュアル発音/消音処理を行ない、一方イベントがなければ、そのままステップ106に進む。

【0073】ステップ106では、LOWERキーボード15bのSWを走査し、続くステップ107にて、このキーボードSWのイベントの有無を判定する。ここでイベントがあれば、ステップ108のLOWERマニュアル発音/消音処理と、ステップ109のLKEYメモリ更新と、ステップ110のDelay Time設定とを行ない、一方イベントがなければ、そのままステップ111に進む。

【0074】ステップ111では、MIDIバッファにMIDI入力があるか否かを調べる。ここでMIDI入力があれば、ステップ112に進み、一方MIDI入力が無ければ、後述するステップ118に進む。ステップ112では、MIDI入力がUPPERキーイベントか否かを調べ、ここでUPPERキーイベントであれば、ステップ113にてUPPERマニュアル発音/消音処理を行なった後にステップ118に進み、一方UPPERキーイベントでなければ、ステップ114にてLOWERキーイベントか否かを調べる。

【0075】ここでLOWERキーイベントであれば、ステップ115のLOWERマニュアル発音/消音処理と、ステップ116のLKEYメモリ更新と、ステップ117のDelay Time設定とを行なう。一方LOWERキーイベントでなければ、ステップ118に進む。

【0076】ステップ118では、BBACCメモリの下8ビットのBEAT値が前回このステップを通った時のBEAT値と違うか否かを調べ、ここで違わなければ前記ステップ101に戻るが、一方違えば変化ありとしてステップ119に進み、DLTIMメモリの値を1減ずる。

【0077】続くステップ120にて、DLTIM

メモリの値をデクリメントした結果を調べ、ここでDLTIM=0になれば、即ちコード検出禁止期間が終了したならば、ステップ121にて、CTYPE/CROOTメモリをチェックしてコードタイプ/コードルートの検出を行なった後に、ステップ122に進み、一方DLTIM=0でなければ後述するステップ300に進む。

【0078】ステップ122では、前記コード検出に基づき、メジャー系かマイナー系かのパート指定を行ないPARTメモリに格納する。そして、後に詳述するステップ300にて自動伴奏の発音/消音処理を行なった後に、ステップ101に戻る。

【0079】つまり、上述したステップ100~300の処理は、操作パネル13やキーボード15を走査するとともにMIDI入力を有無を調べて、それらの入力に応じて発音や消音の処理等を行なうものである。次に、本実施例の要部であるメインルーチン中のステップ200の「パネル処理」について、図12のフローチャートに基づいて説明する。

【0080】ステップ201にて、ONE FINGER SW35又はMEMORY SW343に変化があるか否かを調べ、ここでSWオン変化がある場合には、ステップ202にて、対応LED37を点灯させるとともに、リズムスタイルの変更の処理を行なう。一方、SWオンに変化がない場合には、ステップ203に進む。

【0081】ステップ203では、UPPER/LOWERキーボード15a、15bの音色(尚、前記図3の操作パネル13における音色指定SWは省略してある)のイベントが有るか否か、即ち音色に変化があったか否かを調べ、ここで音色に変化がある場合は、ステップ204にて、その音色を更新し、楽音発生器9に送る。一方音色に変化がない場合は、ステップ205に進む。

【0082】ステップ205では、テンポCKフラグCKFLGが内部設定を示す「INT」であるか否かを調べ、ここで「INT」であれば、ステップ206に進み、一方「INT」でなければ、ステップ209に進む。ステップ206では、テンポ操作子24によるテンポの変更を指示する信号(テンポインクリメント)に変化があったか否かを調べ、ここで変化があれば、ステップ207にてTEMPOメモリの値を更新し、それに基づき操作パネル13のテンポ表示も更新する。そして、このテンポの更新に伴って、上述したDelayTimeの値も変更されることになる。

【0083】ステップ208にてTIMER2のプリセット値を更新し、後述する割込み(INT2)同期を修正する。続くステップ209にて、スタート/ストップフラグSSFLGのビット7「S/S」に変化があったか否かを調べ、ここで変化があれば、スタートイベント若しくはストップイベントのため、ステップ210にて累算小節数BBACCの値をクリアし、一方変化がなけ

れば、ステップ211に進む。

【0084】ステップ211では、TR1ゲートフラグT1GFに変化があったか否かを調べ、ここで変化があったならばステップ212に進み、一方変化がなければステップ221に進む。ステップ212では、変化がON変化(OFF→ON)であったか否かを調べ、ここでON変化でなければ、TR1が発音されなくなったことを意味するので、ステップ213にてT1GF=φとし、ステップ214にて現在発音中のリズム音を消去する。

【0085】一方、ON変化であれば、TR1が新たに(途中から)発音されるようになったことを意味するので、即ち伴奏されるトラックが変更になったので、ステップ215にてTR1ゲートフラグT1GF=1とし、本実施例の要部であるステップ216に進む。

【0086】このステップ216では、異なるループ小節数を有するトラックに変更されたので、次に伴奏される小節を決める処理が行われる。つまり、累算小節数BBACCをTR1のループ小節数TLBで割った余りの小節数を求め、この余りの小節数に基づいて現在のTR1ポインタT1PNTを設定する。即ち余りの小節数に対応して定まるBEAT数を、現在のTR1ポインタT1PNTとする。

【0087】これによって、ループ小節数が異なるトラックに切り換えられた場合でも、容易に次に伴奏すべき小節を求めることができる。続くステップ217にて、リズム発音処理が行われる。つまり、まず現在の小節、BEATにおける設定音量を把握するために、現在の小節の小節マークから読出したVOLUMEをT1VOLに格納し、それ以降現在のBEATまでにVOLUME(音量)データがあれば更にT1VOLUMEを補正する。このT1VOLデータは楽音発生器9に送られる。その後現在のタイミングにて発音開始するリズムがあればそのリズムを発音する。

【0088】そして、続くステップ221~227のTR2処理と、図13のステップ231~237のTR3処理に関しては、TR1処理とほぼ同一のため説明を省く。但し、ステップ227、237のベース/ACC発音処理では、ステップ217のリズム発音処理にはなかった音色ナンバ設定がある。

【0089】つまり、ステップ227では、まず現在の小節、BEATにおける設定音量/音色を把握するために、現在の小節の小節マークから読出したTONE No. (音色)とVOLUME (音量)をT2TON、T2VOLにそれぞれ格納し、それ以降現在のBEATまでにTONE No.データやVOLUMEデータがあれば、更にT2TON、T2VOLを補正している。

【0090】次に、図13のステップ241に進み、このステップでは、(リズムスタイルSW33の操作によって)リズムスタイルに変更されたか否かを調べ、こ

で変化があれば、新たなリズムスタイルNo.をRYM STLメモリに格納するとともに、操作パネル13の対応LED37を点灯する。

【0091】リズムスタイルの変更に伴って行われる続くステップ243では、TR1ゲートフラグT1GF=1か否かを調べ、ここでT1GF=1であれば、ステップ244にて、次に発音すべき小節を決める処理が行われる。つまり、ステップ244では、現在発音中のリズムスタイルを変更し、急に別の(リズムスタイルの)伴奏パターンの中から発音を再開するために、前記ステップ216等と同様にして、累算小節数BBACCを(新しいリズムスタイルの)TR1のループ小節数TLBで割り、その余りの小節数に基づいて発音すべき小節を決める。つまり余りの小節数に対応したBEAT数を現在のTR1ポインタT1PNTとする。

【0092】そしてステップ245では、前記ステップ217で行なった様なリズム発音処理を行なう。続くステップ246では、TR2ゲートフラグT2GF=1か否かを調べ、そうであればステップ247、248にてTR1と同様の処理を行なう。ステップ249では、TR3ゲートフラグT2GF=1か否かを調べ、そうであればステップ250、251にてTR1と同様の処理を行なう。

【0093】尚、ステップ248、251にて、音量ばかりでなく音色に関する処理も行なうのは言うまでもない。つまり、上述したステップ201~251の処理は、操作パネル13に設けられた各種のSWのマニュアル操作に基づいて、伴奏されるループ小節の再生位置を指定する処理等を行なうものである。

【0094】次に、前記メインルーチン中のステップ300の「自動伴奏の発音/消音処理」について、図14のフローチャートに基づいて説明する。このサブルーチンでは、TIMER2割込みで行なう累算小節数、BEAT数の歩進結果に基づき、このタイミングで読出すべきTR1、2、3の自動伴奏があれば読出して発音させる。この処理は、従来はTIMER2の割込処理として行なわれることが多かったが、そうすると割込処理時間が長くなり、時には割込処理中に次の割込みがはいることもあり、その結果メインルーチンでキースキャンがおろそかになることがあるので、この様にメインルーチンに自動伴奏の発音処理を設けた。

【0095】まず、ステップ301にて、TR1ゲートフラグT1GF=1か否かを調べる。ここでT1GF=1であれば、ステップ302にて現在のポイントT1PNTで示される番地から読出したSTEP TIMEをT1STPメモリに格納し、一方T1GF=1でなければ、ステップ311に進む。ステップ303では、前記STEP TIMEと現在のBEATとを比較し、BEATがそのSTEP TIME以上か否かを調べる。ここで肯定判断されると、ステップ304に進み、T1P

NT+1, +2, +3のアドレスからリズムデータを読み出し、ステップ305にて楽音発生器9に対して割当て発音開始指令を送るとともに、その発音にかかるゲートタイムをT1GTEの8CHのうちいずれかに格納する。一方否定判断されると、このループを抜けてステップ307に進む。

【0096】ステップ306では、ポインタを+4して次のリズムデータをアドレスする。その後ステップ302に戻り、次のリズムデータもこのタイミングで発音すべきか否かを順次調べる。そして、(前記ステップ303の判定によって)このタイミングで発音すべきリズムデータを全部読出したら、ステップ307にてリズム8CHのゲートタイムT1GTE_nから1減じる。(但し既にφのものはその必要ない。)

続くステップ308で、ゲートタイムT1GTE_nが(1を減じた結果)φになったか否かを調べ、ここでT1GTE_n=φになれば、そのリズムは発音を終了しているとして、ステップ309にてリズム対応エンベロープをリリースさせ、消音する。一方T1GTE_n=φでなければ、ステップ311に進む。

【0097】ステップ311では、TR2フラグT2GF=1か否かを調べる。ここでT2GF=1であれば、ステップ312にて現在のポイントT2PNTで示される番地から読出されたSTEP TIMEをT2STPメモリに格納し、一方T2GF=1でなければ、図15のステップ321に進む。

【0098】ステップ313では、前記STEP TIMEと現在のBEATとを比較し、BEATがそのSTEP TIME以上か否かを調べる。ここで肯定判断されるとステップ314に進み、一方否定判断されるとステップ318に進む。ステップ314では、T2PNT+2, 3, 4のアドレスからベースデータを読み出し、ステップ315にて、(T2PNT+4)で読出されたPA, PBデータとパート指定データPARTとを比較することで指定パート(メジャー系/マイナー系)のデータであるか否かを調べる。

【0099】ここで指定パートであれば、ステップ316にて、まず読出した音量に、コードタイプCTYPE特定の音名に関するアドバイスが必要ならばその分加算し、その後コードルートCROOT分更に加算する

(尚、その結果で指定音域にない場合はオクターブの移動が必要)。そして、このコードの情報を楽音発生器9に対して割当てると共に、ベース用のゲートタイムT2GTEの2バイトのいずれかに今回のゲートタイムを格納する。一方指定パートでなければ、そのままステップ317に進む。

【0100】ステップ317では、ポインタを+4して次のベースデータをアドレスする。その後ステップ312に戻り、次のベースデータもこのタイミングで発音すべきか否かを順次調べる。そして、(前記ステップ31

3の判定によって)このタイミングで発音すべきリズムデータを全部読出したら、ステップ318にて、ベース2CHのゲートタイムT2GTE_nから1減じる。(但し既にφのものはその必要ない。)

続くステップ319で、ゲートタイムT2GTE_nが(1を減じた結果)φになったか否かを調べ、ここでT2GTE_n=φになれば、そのベースは発音を終了しているとして、ステップ320にてベース対応エンベロープをキーオフ状態とし、消音する。一方T2GTE_n=φでなければ、ステップ321に進む。

【0101】尚、図15のステップ321~330のACC伴奏読出し発音処理に関しては前記ステップ311~320のベース伴奏とほぼ同一なので説明は省略する。但しステップ316ではLKEYメモリに記憶されている押鍵情報(最大8)全てに対し、読出したタイミングとベロシティにて同時に楽音発生器9に割当て発音させる。その時ゲートタイムの管理は、全て同じため1つで済む。

【0102】次に、INT1の割込ルーチン、即ちMIDI入力時の割込ルーチンについて、図16のフローチャートに基づいて説明する。この割込みルーチンは、MIDI入力情報に対応してテンポスピードを設定する部分であり、特にキーコードに関してはMIDI INキーバッファへ格納する。また、タイミングCKに関しては、テンポCKフラグCKFLGが外部設定を示す「EXT」と指定されている時に限り、その入力タイミングCKの周期から算出した値で、後述する図17の様にTIMER2を動作させる。

【0103】まず、ステップ401にてMIDI入力がキー情報か否かを調べ、ここでキー情報であれば、ステップ402にてそのキー情報をMIDI INキーバッファへ格納する。一方、キー情報でなければ、ステップ403にて、MIDI入力がタイミングCKか否かを調べ、ここでタイミングCKであれば、ステップ404にてテンポCKフラグCKFLG=EXTかを調べる。つまり、テンポの外部設定であるか否かを確認する。

【0104】ここで外部設定であれば、ステップ405にて前回のタイミングCKから今回までの時間をTIMER1から読出し、前回と今回とのタイミングCKの時間差(周期)CKCNTを求め、CKCNTメモリに記憶する。続く407にて、CKCNTメモリに記憶した今回の時間差を含む最新の4回の時間差の平均CKAVRNを求め、CKAVRNメモリへ格納する。

【0105】続くステップ408にて、現在のテンポスピードを形成しているCKAVROと前記最新のCKAVRNとを比べ、その差が一定以上か否かを調べる。ここで差が一定以上であれば、テンポスピードを更新するために、ステップ409にてCKAVRNの値をCKAVROへ代入し、ステップ410にてCKAVROに基づきテンポ表示を更新するとともに、ステップ411に

てTIMER 2へ更新したCKAVROに基づくプリセット値を入れ、ステップ412に進む。一方、差が一定以上でなければそのままステップ412に進む。

【0106】ステップ412では、一定時間タイミングCKの入力がないか否かを調べ、ここで入力がない場合は、ステップ413にてTIMER 2をホールドして、一旦本処理を終了する。つまり、このステップ401～413の処理では、MIDI入力のうちタイミングCKの入力があつた場合には、外部設定であることを確認して、前回と今回とのタイミングCKの時間差を求め、更にその最新の4回の平均を求めている。そして、所定の条件下でこの最新の平均を用いて、テンポの更新を行なっているので、テンポが変動した場合も適切なテンポ設定が可能である。

【0107】次に、INT 2の割込ルーチン、即ちTIMER 2のテンポ割込ルーチンについて、図17のフローチャートに基づいて説明する。この割込みルーチンは、設定されたテンポスピードに応じて実際に伴奏のスピードを規定するためのもので、TIMER 2のテンポスピードに対応したプリセッタブルダウンカウンタがゼロになったときに発生する。

【0108】まず、ステップ501にて、スタート/ストップフラグSSFLGのビット7 (S/S) = 1か否かを調べる。ここで S/S = 1であれば、ステップ502にて累算小節数BBACCを+1し、一方 S/S = 1でなければ、一旦本処理を終了する。

【0109】ステップ503では、拍子が3/4か4/4かを調べ、3/4であればステップ504に進み、4/4であればステップ507に進む。ステップ504では、3/4拍子であるので、BEATが144以上か否かを調べる。ここで144以上であれば、ステップ505にてBEATから144を引き、ステップ506にてBARを+1する。

【0110】一方、ステップ507では、4/4拍子であるので、BEATが192以上か否かを調べる。ここで192以上であればステップ508にてBEATから192を引き、ステップ509にてBARを+1する。そして、続くステップ510にて、前記BAR、BEATを10進に変更して表示し、一旦本処理を終了する。

【0111】つまり、この処理によって、設定されたテンポにて、順次発音のための処理が実施されることになる。以上詳述した様に、本実施例の電子オルガンでは、各トラックに記憶された(リズムやベース等の伴奏データの)ループ小節数が異なっている場合でも、累算小節数BBACCを各ループ小節数で割った余りの小節数に基づいて、次に伴奏すべき小節の決めることができる。よって、ある伴奏の途中でリズムスタイルが切り換えられた場合やトラックが切り換えられた場合でも、迅速かつ容易に次に伴奏すべき小節を決めることができるので、違和感のない伴奏を行なうことができる。

【0112】しかも、本実施例では、リズムスタイル毎に各トラックに記憶するループ小節数を最小にすることができるので、伴奏の質を低下させることなく、メモリの使用領域を低減できるという顕著な効果がある。尚、本発明は、上記実施例に何等限定されず、本発明の要旨の範囲内において各種の態様で実施できることは勿論である。

【0113】例えば他の実施例として、前記ステップ216、244等における様に、次に伴奏される小節を決めるために、除算によって余りの小節数を求める手順ではなく、(2進数で示される)BARカウンタ(表1参照)の上位ビットにマスクを施す処理を行なってもよい。

【0114】具体的には、例えばTR 1が4小節ループである場合には、BBACCメモリに記憶されている現在のBARカウンタ(図10及び表1参照)と、上述した表2に示す4小節ループに相当するマスクデータとの論理積を求め、この論理積に該当する小節を次に伴奏される小節とする。そして、求められた小節に基づいて、現在のTR 1ポインタTPNTを設定する。

【0115】この様に、本実施例では、前記実施例と同様な効果を奏するとともに、リズムスタイル及びトラックが切り換えられた際には、BARカウンタにマスクをかけることによって、即ちBARカウンタとマスクデータとの論理積を求めることによって、たとえリズムスタイルやトラック毎にループ小節数が異なっている場合でも、一層迅速かつ容易に次に伴奏すべき小節を決めることができる。

【0116】

【発明の効果】以上詳述した様に、本発明の自動演奏装置では、ループして演奏される演奏データの累積小節数を、ループ小節数で割った余りに基づいて再生小節を求めることができる。即ち、記憶されたループ小節数が異なっている場合でも、次に演奏すべき小節がループ小節の何番目かを求めることができるので、この余りに基づいて容易に演奏データの再生小節を指示することができる。従って、この再生小節から演奏を再生すると、リズムスタイルが切り換えられた場合や途中から演奏が開始される場合などでも、違和感のない演奏行なうことができる。

【0117】また、この様に、例えば各リズムスタイルや各トラック毎に、演奏に用いられる最小のループ小節数を違えて別個に設定することができるので、演奏の質を低下させることなく、メモリを低減できるという顕著な効果を奏する。特に、各ループ小節数を2の累乗とし、2進数で表示される累算小節数の所定ビットをマスクすると、迅速にかつ容易に再生小節が分かるので好適である。

【図面の簡単な説明】

【図1】 請求項1の発明の構成を例示する概略構成図である。

【図2】 実施例の電子オルガンの電氣的構成を示すブロック図である。

【図3】 電子オルガンの操作パネルを示す正面図である。

【図4】 電子オルガンのヘッダーの並びを示す説明図である。

【図5】 各データの小節の構造を示す説明図である。

【図6】 各小節内のデータの並びを示す説明図である。

【図7】 各伴奏データ及び小節マークの構造を示す説明図である。

【図8】 オートペンダを示し、(a)はその変化の状態を示すグラフ、(b)はそのデータの構造を示す説明図である。

【図9】 RAMの各種レジスタを示す説明図である。

【図10】 レジスタに格納されたデータの構造を示す説明図である。

【図11】 CPUのメインルーチンを示すフローチャートである。

【図12】 パネル処理の一部を示すフローチャートである。

【図13】 パネル処理の一部を示すフローチャートである。

【図14】 自動伴奏発音/消音処理の一部を示すフローチャートである。

【図15】 自動伴奏発音/消音処理の一部を示すフローチャートである。

【図16】 MIDI入力の割込処理を示すフローチャートである。

【図17】 TIMER 2のテンポ割込処理を示すフローチャートである。

【図18】 従来技術を示す説明図である。

【符号の説明】

M1…演奏データ記憶手段

M2…ループ演奏手段

M3…累算小節数計算手段

M4…再生小節指示手段

1…CPU

13…操作パネル

32…ゲートスイッチ

イルスイッチ

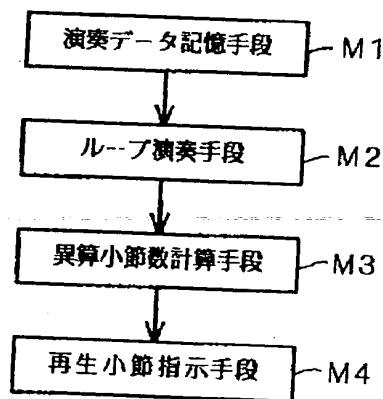
37…LED

7…RAM

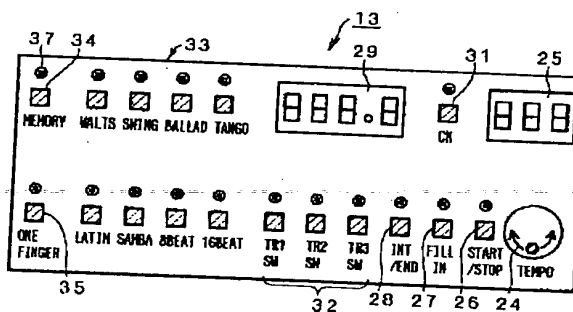
15…キーボード

33…リズムスタ

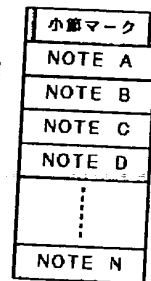
【図1】



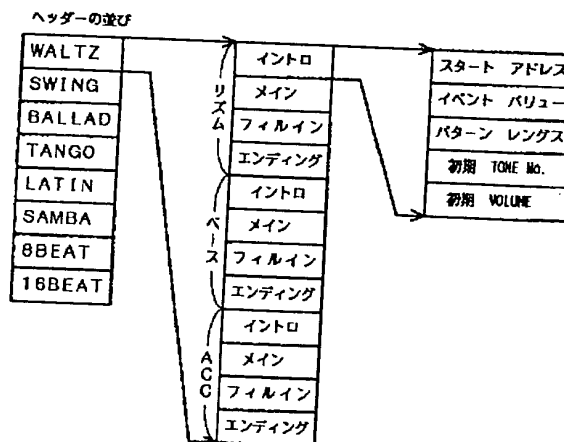
【図3】



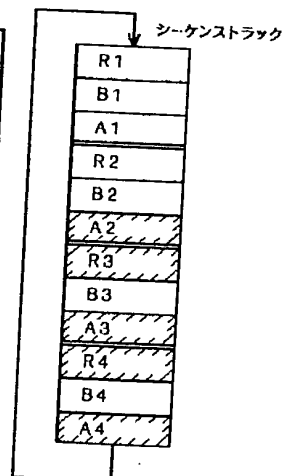
【図6】



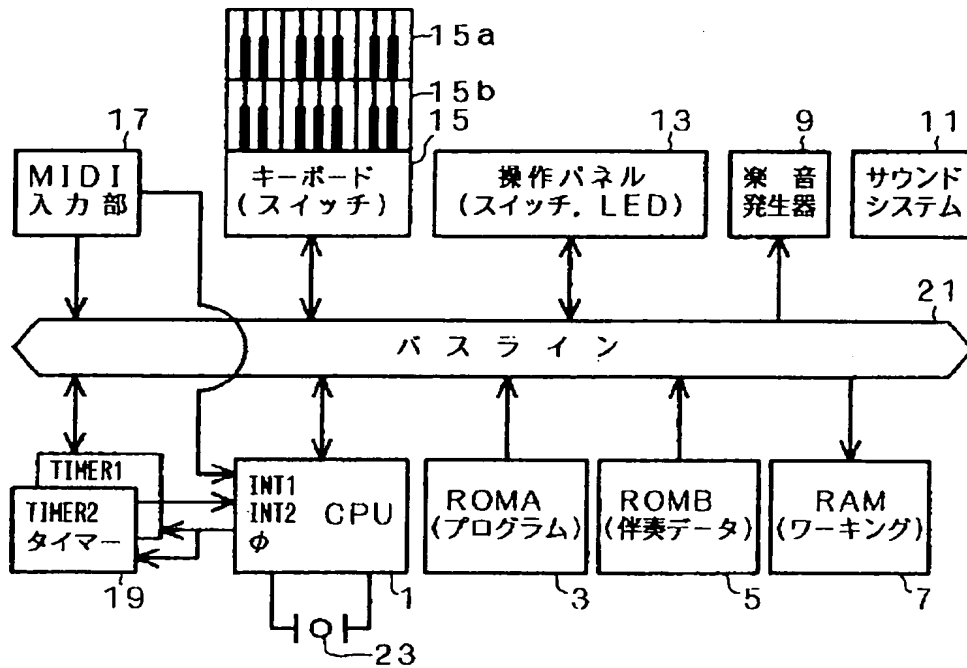
【図4】



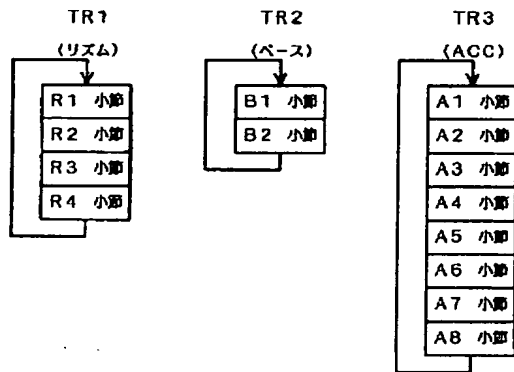
【図18】



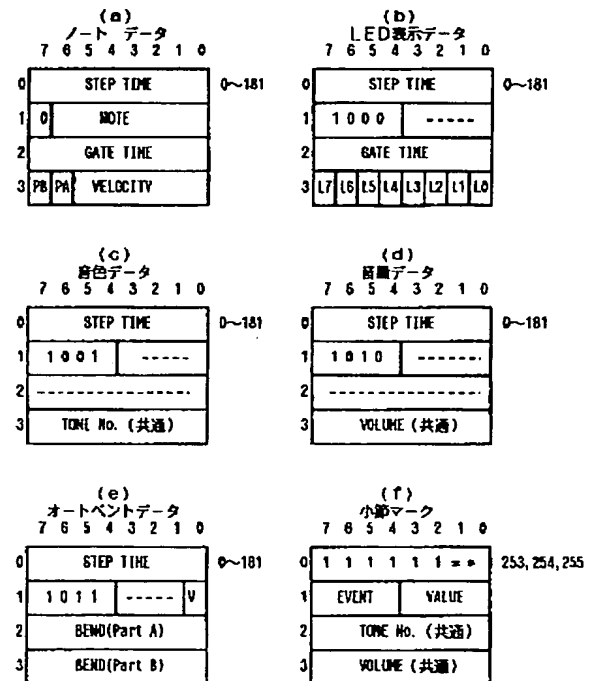
【図2】



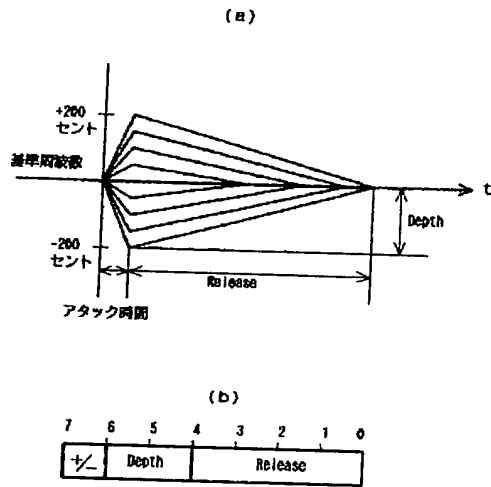
【図5】



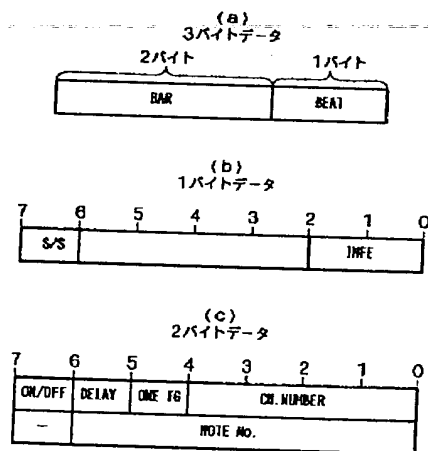
【図7】



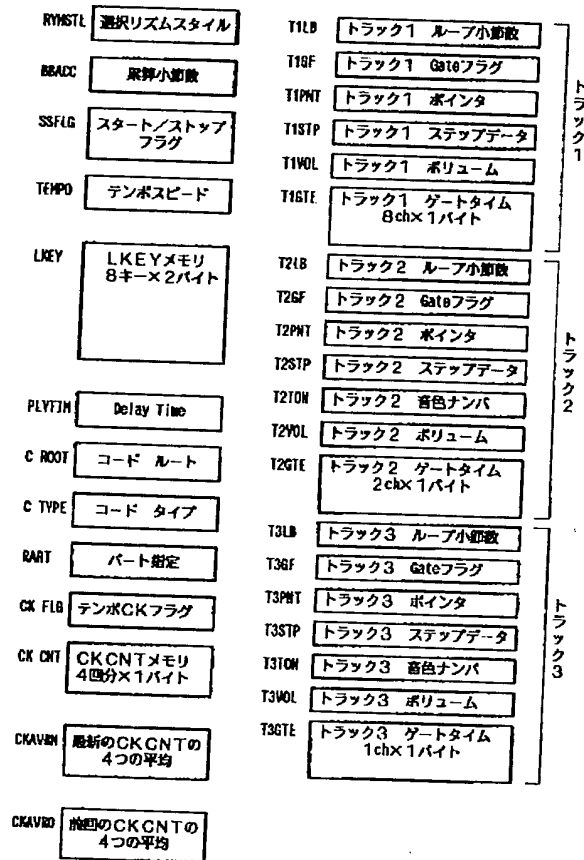
【図8】



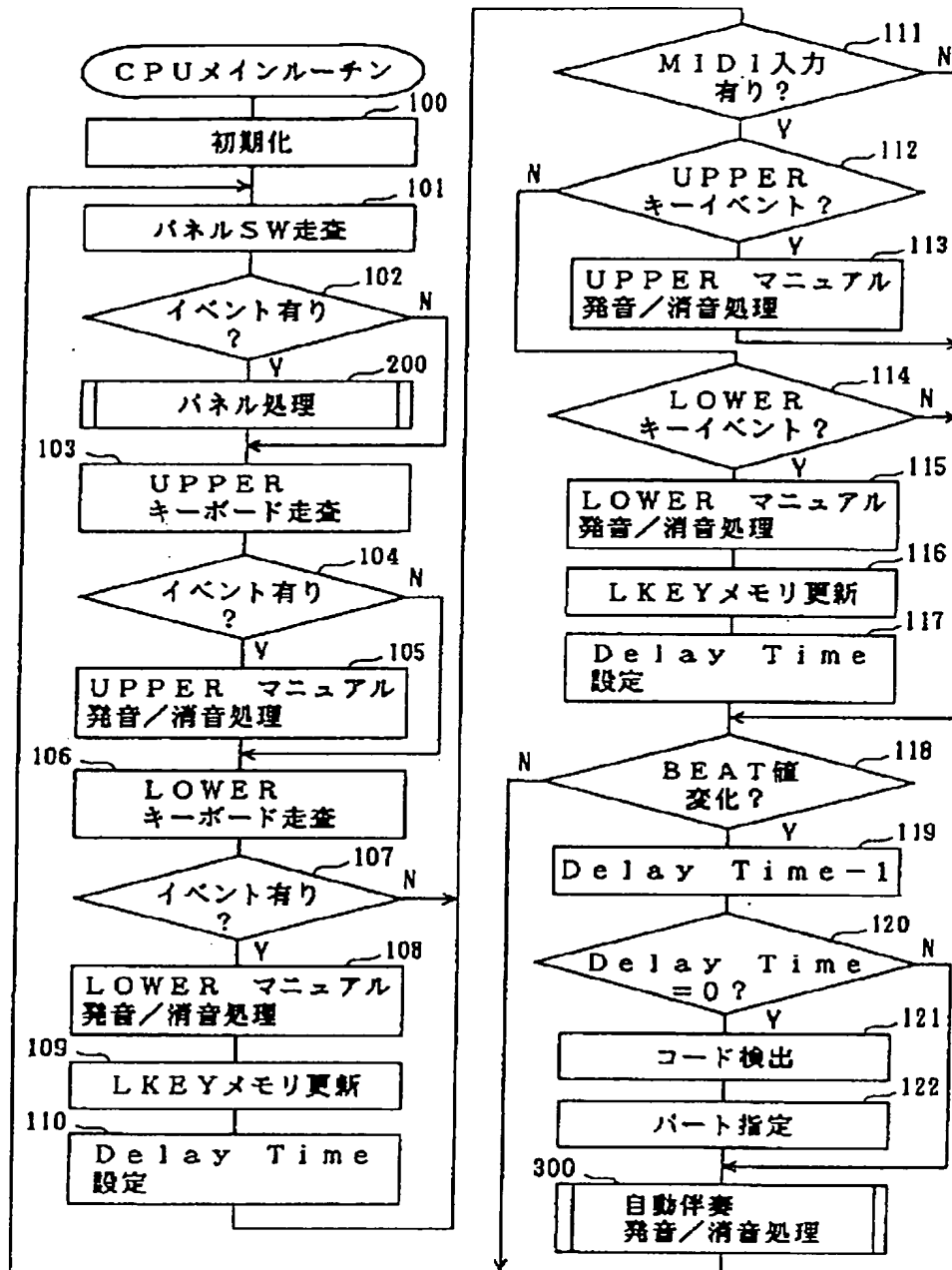
【図10】



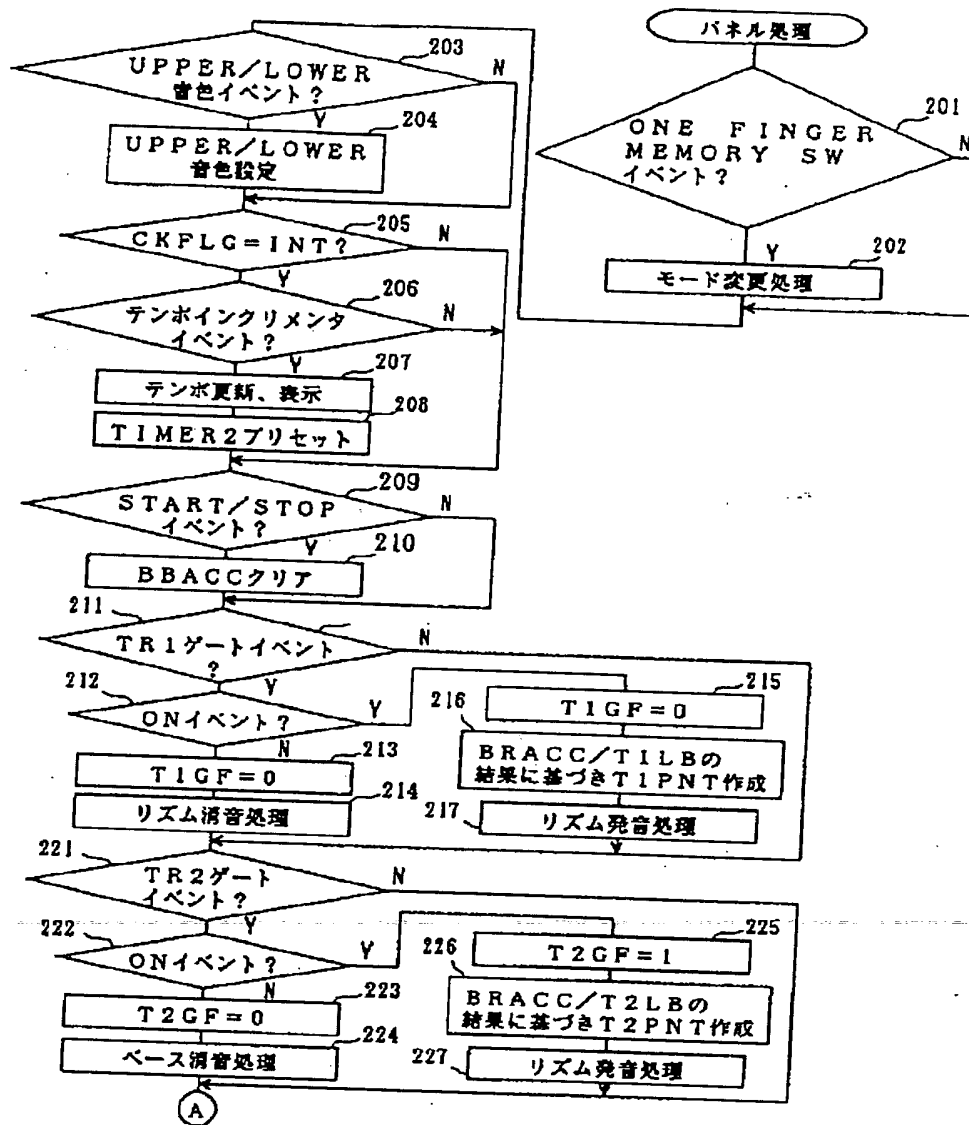
【図9】



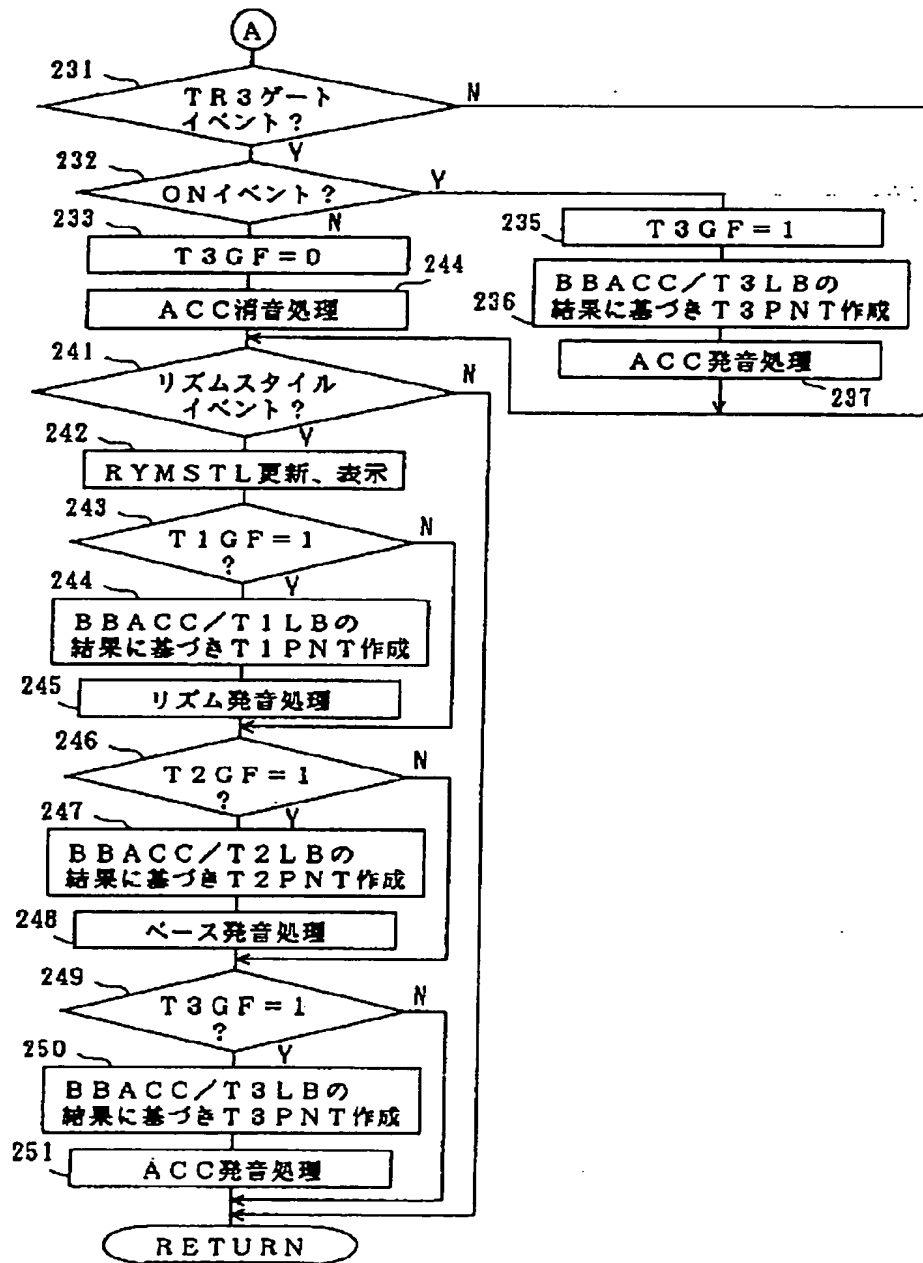
【図11】



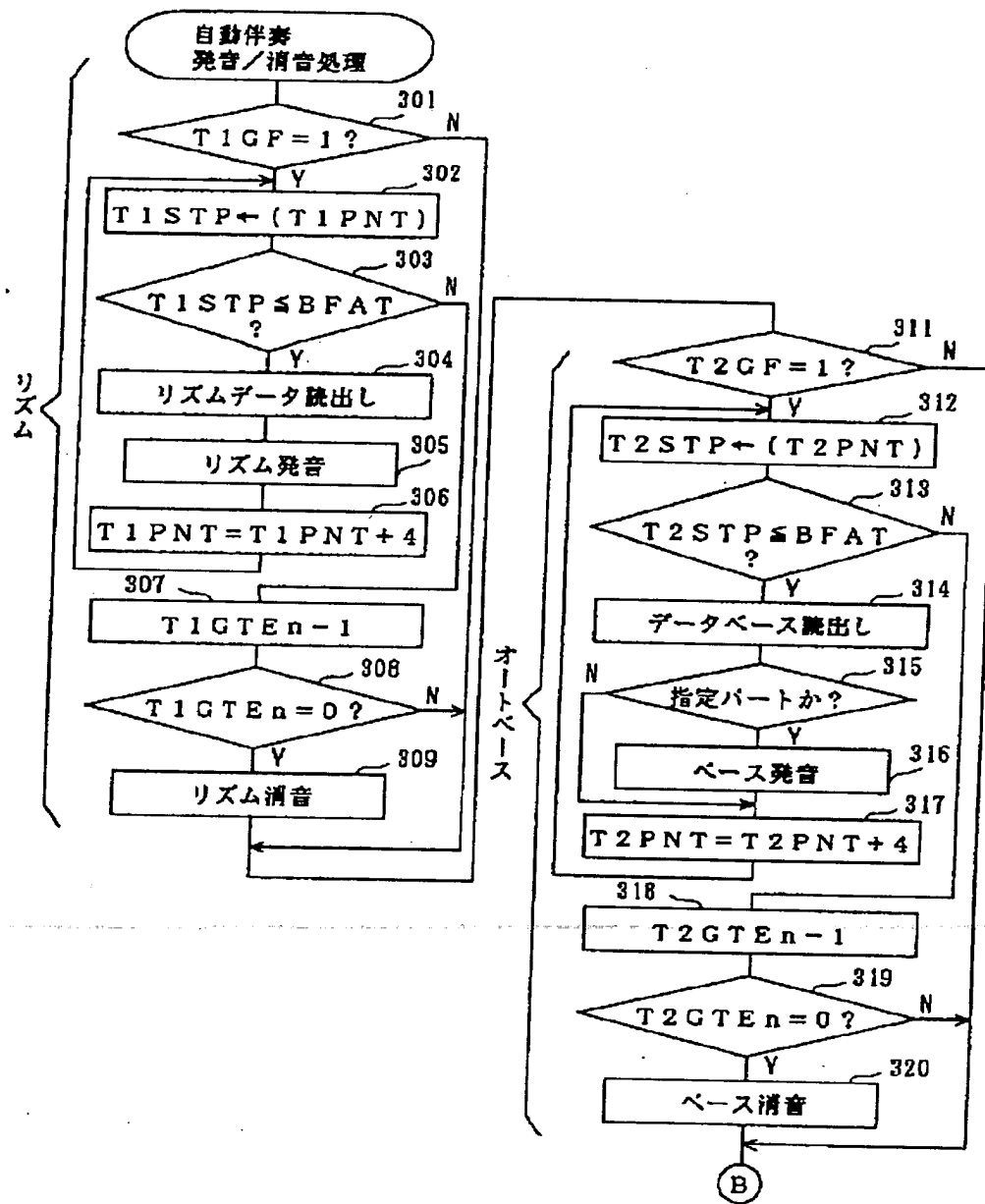
【図12】



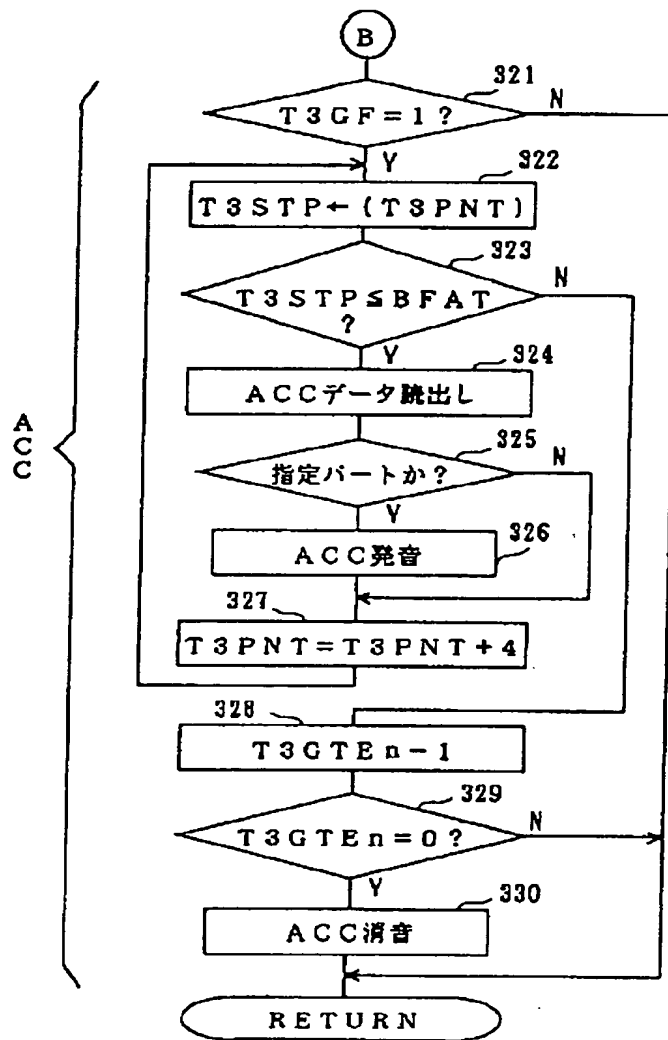
【図13】



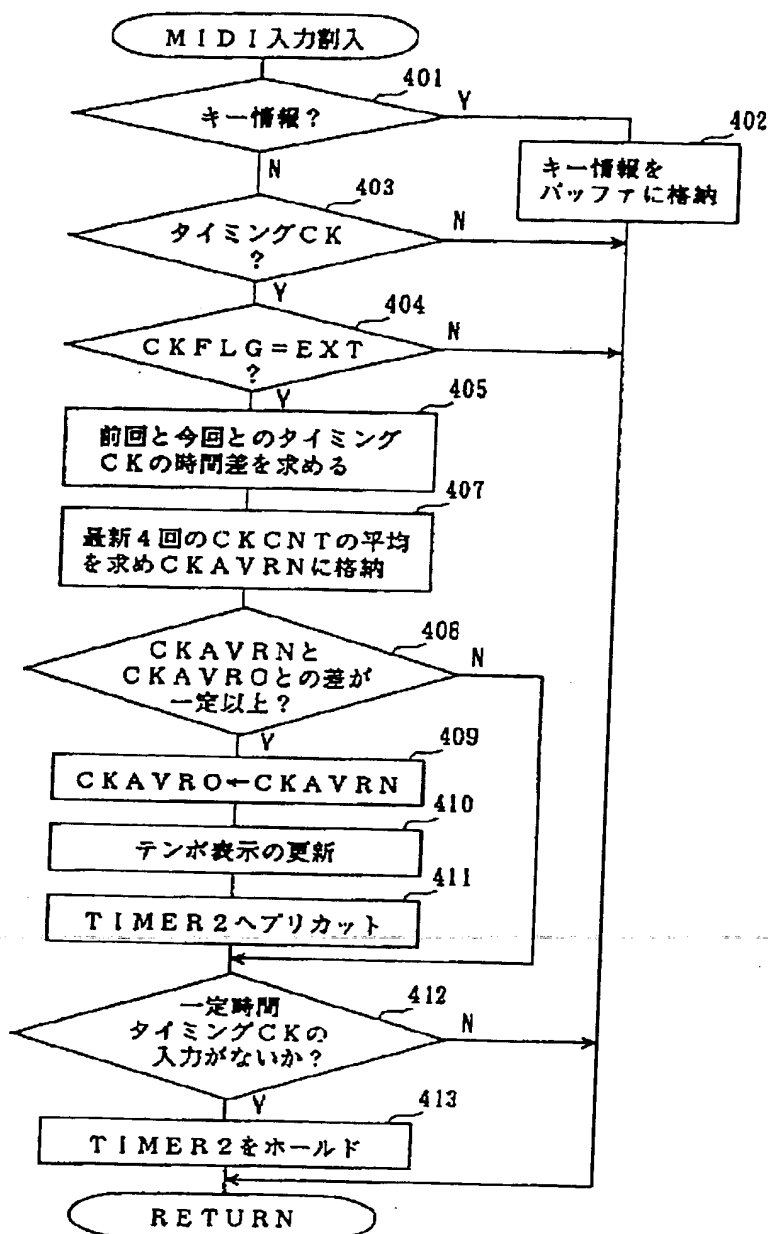
【図14】



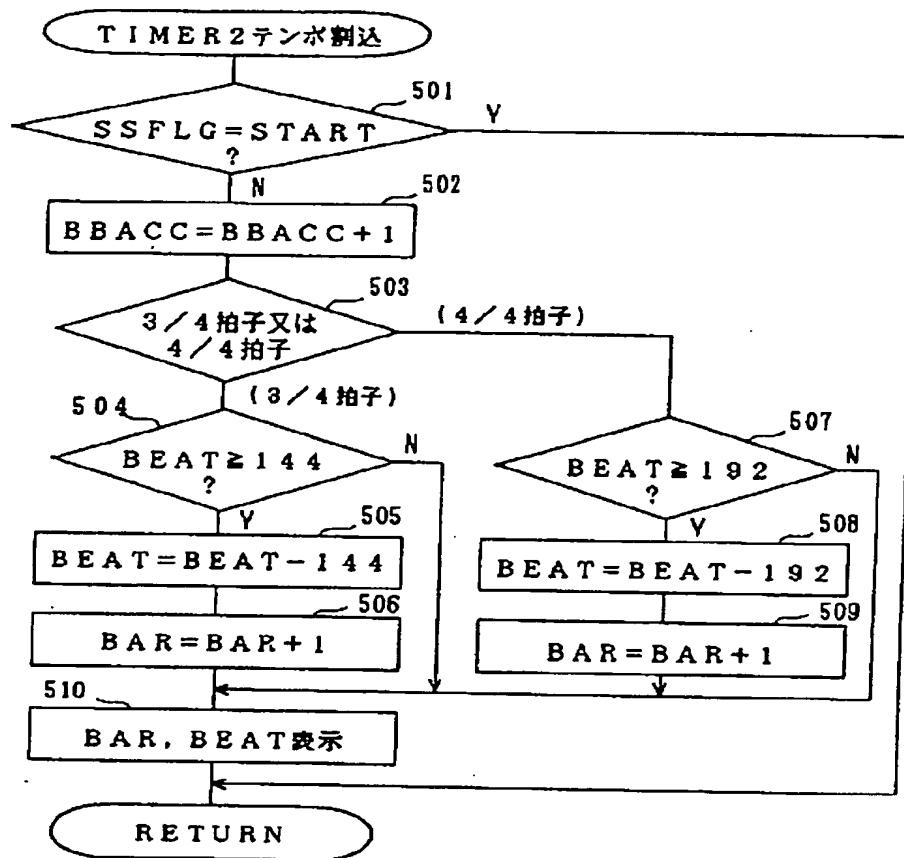
【図15】



【図16】



【図17】



フロントページの続き

(72)発明者 中川 博信
 静岡県浜松市寺島町200番地 株式会社河
 合楽器製作所内

【公報種別】特許法第17条の2の規定による補正の掲載
【部門区分】第6部門第2区分
【発行日】平成10年(1998)7月31日

【公開番号】特開平6—289875
【公開日】平成6年(1994)10月18日
【年通号数】公開特許公報6—2899
【出願番号】特願平5—72365
【国際特許分類第6版】

G10H 1/36
1/00 102
1/40

【FI】

G10H 1/36
1/00 102 Z
1/40

【手続補正書】

【提出日】平成8年11月6日

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】請求項1

【補正方法】変更

【補正内容】

【請求項1】 ループ小節数の異なる複数の演奏データを記憶する演奏データ記憶手段と、
前記演奏データのいずれかを指示する演奏データ指示手段と、

該演奏データ指示手段によって指示された前記演奏データを、前記演奏データ記憶手段から繰り返して読み出して演奏を行なうループ演奏手段と、
該ループ演奏手段による演奏の開始から、演奏される小節数を累算する累算小節数計算手段と、
該累算小節数計算手段によって累算された累算小節数を、前記演奏データ指示手段によって指示された前記演奏データのループ小節数で除した余りに基づいて前記演奏データの再生小節を指示する再生小節指示手段と、
を備えたことを特徴とする自動演奏装置。

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0008

【補正方法】変更

【補正内容】

【0008】

【課題を解決するための手段】この目的を達成するための請求項1の発明は、図1に例示する様に、ループ小節数の異なる複数の演奏データを記憶する演奏データ記憶手段M1と、前記演奏データのいずれかを指示する演奏データ指示手段M2と、該演奏データ指示手段M2によって指示された前記演奏データを、前記演奏データ記憶

手段M1から繰り返して読み出して演奏を行なうループ演奏手段M3と、該ループ演奏手段M3による演奏の開始から、演奏される小節数を累算する累算小節数計算手段M4と、該累算小節数計算手段M4によって累算された累算小節数を、前記演奏データ指示手段M2によって指示された前記演奏データのループ小節数で除した余りに基づいて前記演奏データの再生小節を指示する再生小節指示手段M5と、を備えたことを特徴とする自動演奏装置を要旨とする。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0017

【補正方法】変更

【補正内容】

【0017】

【作用】(1)請求項1の発明では、演奏データ記憶手段M1によって、ループ小節数の異なる複数の演奏データを記憶する。そして、演奏データ指示手段M2によって演奏データのいずれかが指示された場合には、ループ演奏手段M3によって、この指示されたループ小節の演奏データを繰り返して読み出して演奏を行なう。それとともに、累算小節数計算手段M4によって、演奏の開始からの演奏される小節数の累算を行ない、再生小節指示手段M5によって、この累算された累算小節数を、指示された演奏データのループ小節数で除した余りに基づいて演奏データの再生小節を指示する。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0018

【補正方法】変更

【補正内容】

【0018】つまり、ループして演奏される演奏データ

の累積小節数を、演奏データ指示手段M2によって指示された演奏データのループ小節数で割った余りが、再生される小節の位置を示すことになる。これにより、ループ小節数が異なっている場合でも、即座に際すべき小節を指定できるので、例えばリズムスタイルが切り換えられた場合や（最初は再生されなくて）途中からあるパートが再生される場合でも、違和感のない演奏を行なうことが可能である。

【手続補正5】

【補正対象書類名】明細書

【補正対象項目名】0116

【補正方法】変更

【補正内容】

【0116】

【発明の効果】以上詳述した様に、本発明の自動演奏装置では、ループして演奏される演奏データの累積小節数

を、指示された演奏データのループ小節数で割った余りに基づいて再生小節を求めることができる。即ち、記憶された演奏データのループ小節数が異なっている場合でも、次に演奏すべき小節がループ小節の何番目かを求めることができるので、この余りに基づいて容易に演奏データの再生小節を指示することができる。従って、この再生小節から演奏を再生すると、リズムスタイルが切り換えられた場合や途中から演奏が開始される場合などでも、違和感のない演奏を行なうことができる。

【手続補正6】

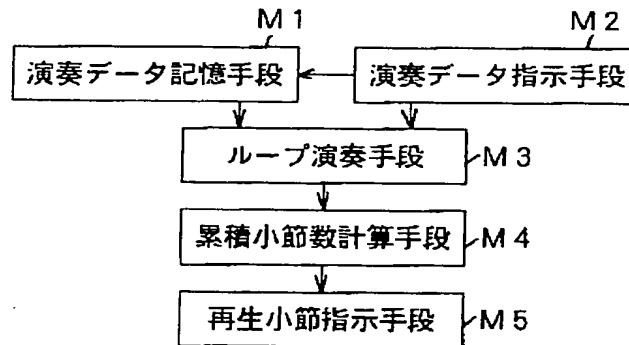
【補正対象書類名】図面

【補正対象項目名】図1

【補正方法】変更

【補正内容】

【図1】



BLANK PAGE

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)